

Amendments to the Claims

This listing of claims will replace all prior versions, and listings, of claims in the application.

1. (Currently Amended) A method comprising:
calculating a Levenshtein matrix of a first string and a second string;
determining a Levenshtein distance from said Levenshtein matrix;
determining a largest common substring from said Levenshtein matrix; and
~~determining a similarity between said first string and said second string as a function of~~
~~said Levenshtein distance and said largest common substring; and~~
~~determining if said first string is a match to said second string based upon said similarity~~
automating data entry, processing or reporting for a database including at least one of
said first or second strings based upon said Levenshtein distance and said largest common
substring.

2. (Original) The method according to Claim 1, wherein determining a largest common substring from said Levenshtein distance matrix comprises determining a longest diagonal of equal hamming distances of a lowest value.

3. (Original) The method according to Claim 1, further comprising calculating a Levenshtein score.

4. (Original) The method according to Claim 1, further comprising determining the length of the largest common substring.

5. (Original) The method according to Claim 4, further comprising calculating a largest common substring score.

6. (Currently Amended) A method comprising:

- calculating a Levenshtein matrix of a first string and a second string;
- determining a Levenshtein distance from said Levenshtein matrix;
- determining a largest common substring from said Levenshtein matrix;
- calculating a Levenshtein score as a function of said Levenshtein distance;
- calculating a largest common substring score as a function of said largest common substring;
- determining a similarity between said first string and said second string as a function of said Levenshtein score and said largest common substring score; and
- ~~determining if said first string is a match to said second string based upon said similarity~~

automating data entry, processing or reporting for a database including at least one of said first or second strings based upon said similarity.

7. (Previously Presented) The method according to Claim 6, further comprising calculating an acronym score of said first string and said second string.

8. (Original) The method according to Claim 7, further comprising calculating a weighted acronym score comprising a product of said acronym score and an acronym weight factor.

9. (Original) The method according to Claim 6, further comprising:
calculating a weighted Levenshtein score comprising a product of said Levenshtein score and a Levenshtein weight factor;
calculating a weighted largest common substring score comprising a product of said largest common substring score and a largest common substring weight factor; and
calculating a Levenshtein/largest common substring score comprising a sum of said weighted Levenshtein score and said weighted largest common substring score.

10. (Original) The method according to Claim 9, wherein a sum of said Levenshtein weight factor and said largest common substring weight factor is equal to one.

11. (Original) The method according to Claim 9, further comprising calculating a first weighted numerical score comprising a product of said Levenstein/largest common substring score and a string weight factor.

12. (Previously Presented) The method according to Claim 11, further comprising:
calculating an acronym score of said first string and said second string;
calculating a weighted acronym score comprising a product of said acronym score and an acronym weight factor; and
calculating a second weighted numerical score comprising a sum of said first weighted numerical score and said weighted acronym score.

13. (Original) The method according to Claim 12, wherein a sum of said string weight factor and said acronym weight factor is equal to one.

14. (Currently Amended) A computer-readable medium containing one or more sequences of instructions which when executed by a computing device cause the computing device to implement a method for determining a similarity comprising:
calculating a Levenshtein matrix of a first string and a second string;
determining a Levenshtein distance from said Levenshtein matrix;
determining a largest common substring from said Levenshtein matrix;
calculating a Levenshtein score as a function of said Levenshtein distance;

calculating a largest common substring score as a function of said largest common substring;

calculating a first numerical score as a function of said Levenshtein score and said largest common substring score, ~~wherein said first numerical score is a first quantification of a similarity between a set of characters of said first string and a set of characters of said second string; and~~

~~determining if said first string is a match to said second string based upon said first numerical score~~

automating data entry, processing or reporting for a database including at least one of said first or second strings based upon said first numerical score.

15. (Previously Presented) The computer-readable medium according to Claim 14, wherein calculating said Levenshtein score comprises:

subtracting the resultant of dividing said Levenshtein distance by an average of a length of said first string and a length of said second string from one.

16. (Previously Presented) The computer-readable medium according to Claim 14, wherein calculating said largest common substring score comprises:

determining a length of said largest common substring from said Levenshtein matrix; and

dividing said length of said largest common substring by an average of a length of said first string and a length of said second string.

17. (Original) The computer-readable medium according to Claim 14, wherein calculating said first numerical score comprises:

calculating a weighted Levenshtein score comprising a product of said Levenshtein score and a Levenshtein weight factor;

calculating a weighted largest common substring score comprising a product of said largest common substring score and a largest common substring weight factor; and

summing said weighted Levenshtein score and said weighted largest common substring score.

18. (Currently Amended) The computer-readable medium according to Claim 14, further comprising:

calculating an acronym score;

calculating a second numerical score as a function of said first numerical score and said acronym score; and

further ~~determining if said first string is a match to said second string~~ automating data entry, processing or reporting based upon said second numerical score.

19. (Original) The computer-readable medium according to Claim 18, wherein calculating said second numerical score comprises:

calculating a weighted Levenshtein score comprising a product of said Levenshtein score and a Levenshtein weight factor;

calculating a weighted largest common substring score comprising a product of said largest common substring score and a largest common substring weight factor;

calculating a Levenshtein/largest common substring score comprising a sum of said weighted Levenshtein score and said weighted largest common substring score;

calculating a weighted Levenshtein/largest common substring score comprising a product of said Levenshtein/largest common substring score and a Levenshtein/largest common substring weight factor;

calculating a weighted acronym score comprising a product of said acronym score and an acronym score weight factor; and

summing said weighted Levenshtein/largest common substring score and said weighted acronym score.

20. (Currently Amended) The computer-readable medium according to Claim 19, further comprising:

utilizing said first numerical score for ~~determining said similarity~~ automating data entry, processing or reporting, when said first string and said second string comprise numerical-type strings; and

utilizing said second numerical score for ~~determining said similarity~~ automating data entry, processing or reporting, when said first string or said second string comprise character-type strings.